On Isolating Points using Disks

Matt Gibson[1], Gaurav Kanade[2], and Kasturi Varadarajan[2]

[1] Department of Electrical and Computer Engineering
The University of Iowa
Iowa City, IA 52246
Email:matthew-gibson@uiowa.edu
[2] Department of Computer Science
The University of Iowa
Iowa City, IA 52246
Email:gaurav-kanade@uiowa.edu, kvaradar@iowa.uiowa.edu

Abstract. In this paper, we consider the problem of choosing disks (that we can think of as corresponding to wireless sensors) so that given a set of input points in the plane, there exists no path between any pair of these points that is not intercepted by some disk. We try to achieve this separation using a minimum number of a given set of unit disks. We show that a constant factor approximation to this problem can be found in polynomial time using a greedy algorithm. To the best of our knowledge we are the first to study this optimization problem.

1 Introduction

Wireless sensors are being extensively used in applications to provide barriers as a defense mechanism against intruders at important buildings, estates, national borders etc. Monitoring the area of interest by this type of coverage is called *barrier* coverage [11]. Such sensors are also being used to detect and track moving objects such as animals in national parks, enemies in a battlefield, forest fires, crop diseases etc. In such applications it might be prohibitively expensive to attain blanket coverage but sufficient to ensure that the object under consideration cannot travel too far before it is detected. Such a coverage is called *trap* coverage [3,14].

Inspired by such applications, we consider the problem of isolating a set of points by a minimum-size subset of a given subset of unit radius disks. A unit disk crudely models the region sensed by a sensor, and the work reported here readily generalizes to disks of arbitrary, different radii.

Problem Formulation. The input to our problem is a set I of n unit disks, and a set P of k points such that I *separates* P, that is, for any two points $p, q \in P$, every path between p and q intersects at least one disk in I. The goal is to find a minimum cardinality subset of I that separates P. See Figure 1 for an illustration of this notion of separation.

There has been a lot of recent interest on geometric variants of well-known NP-hard combinatorial optimization problems, and our work should be seen in
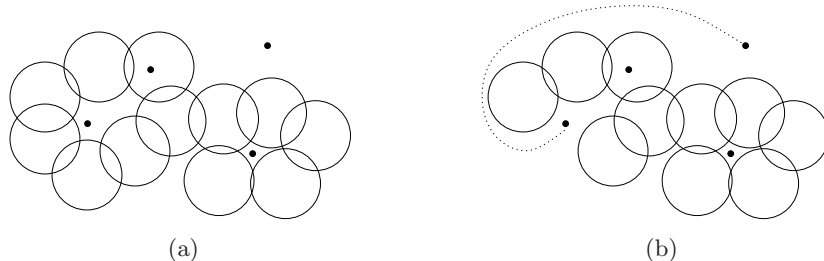
Fig. 1. (a) This set of disks separates the points because every path connecting any two points must intersect a disk. (b) This set of disks does not separate the points.

this context. For several variants of the geometric set cover problem, for example, approximation algorithms have been designed [7, 2, 12] that improve upon the best guarantees for the combinatorial set cover problem. For the problem of covering points by the smallest subset of a given set of unit disks, we have approximation algorithms that guarantee an $O(1)$ approximation and even a PTAS [4, 12]. These results hold even for disks of arbitrary radii. Our problem can be viewed as a set cover problem where the elements that need to be covered are not points, but paths. However, known results only imply a trivial $O(n)$ approximation when viewed through this set cover lens.

Another example of a problem that has received such attention is the *independent set* problem. For many geometric variants [5, 6, 9], approximation ratios that are better than that for the combinatorial case are known.

Our problem is similar to the node multi-terminal cut problem in graphs [10]. Here, we are given a graph $G = (V, E)$ with costs on the vertices and a subset $U \subseteq V$ of k vertices, and our goal is to compute a minimum cost subset of vertices whose removal disconnects every pair of vertices in U. This problem admits a poly-time algorithm that guarantees an $O(1)$ approximation. We note however that the problem we consider does not seem to be a special case of the multi-terminal cut problem.

Contribution and Related Work. Our main result is a polynomial time algorithm that guarantees an $O(1)$ approximation for the problem. To the best of our knowledge, this is the first non-trivial approximation algorithm for this problem. Our algorithm is simple and combinatorial and is in fact a greedy algorithm. We first present an $O(1)$ approximation for the following two-point separation problem. We are given a set of unit disks G, and two points s and t, and we wish to find the smallest subset $B \subseteq G$ so that B separates s and t.

Our greedy algorithm to the overall problem applies the two-point separation algorithm to find the cheapest subset B of I that separates some pair of points in P. Suppose that P is partitioned into sets $P_1, P_2, \ldots, P_\tau$ where each P_i is the subset of points in the same "face" with respect to B. The algorithm then recursively finds a separator for each of the P_i, and returns the union of these and B.

The analysis to show that this algorithm has the $O(1)$ approximation guarantee relies on the combinatorial complexity of the boundary of the union of disks. It uses a subtle and global argument to bound the total size of all the separators B computed in each of the recursive calls.[3]

Our approximation algorithm for the two-point separation problem, which is a subroutine we use in the overall algorithm, is similar to fast algorithms for finding minimum s-t cuts in undirected planar graphs, see for example [13]. Our overall greedy algorithm has some resemblance to the algorithm of Erickson and Har-Peled [8] employed in the context of approximating the minimum cut graph of a polyhedral manifold. The details of the our algorithm and the analysis, however, are quite different from these papers since we do not have an embedded graph but rather a system of unit disks. Sankararaman et al. [14] investigate a notion of coverage which they call *weak coverage*. Given a region $\mathcal{R}$ of interest (which they take to be a square in the plane) and a set I of unit disks (sensors), the region is said to be k-weakly covered if each connected component of $\mathcal{R} - \bigcup_{d \in I} d$ has diameter at most k. They consider the situation when a given set I of unit disks *completely* covers $\mathcal{R}$, and address the problem of partitioning I into as many subsets as possible so that $\mathcal{R}$ is k-weakly covered by every subset. Their work differs in flavor from ours mainly due to the assumption that I completely covers $\mathcal{R}$.

Organization. In Section 2, we discuss standard notions we require, and then reduce our problem to the case where none of the points in input P are contained in any of the input disks. In Section 3, we present our approximation algorithm for separating two points. In Section 4, we describe our main result, the constant factor approximation algorithm for separating P. We conclude in Section 5 with some remarks.

2 Preliminaries

We will refer to the standard notions of vertices, edges, and faces in arrangements of circles [1]. In particular, for a set R of m disks, we are interested in the faces in the complement of the union of the disks in R. These are the connected components of the set $\Re^2 - \bigcup_{d \in R} d$. We also need the combinatorial result that the number of these faces is $O(m)$. Furthermore, the total number of vertices and edges on the boundaries of all these faces, that is, the combinatorial complexity of the boundary of the union of disks in R, is $O(m)$ [1]. We make standard general position assumptions about the input set I of disks in this article. This helps simplify the exposition and is without loss of generality.

---

[3] In the earlier version of this paper, a similar algorithm was analyzed in a more "local" fashion. The basic observation was that the very first separator B that is computed has size $O(|OPT|/k)$, where OPT is the optimal solution for the problem. Subsequent separators computed in the recursive calls may be more expensive, but it was shown that the overall size is $O((\log k) \cdot |OPT|)$. In contrast, the present analysis does not try to bound the size of the individual separators, but just the sum of their sizes. As a consequence, the analysis also turns out to be technically simpler.

Lemma 1. *Let R be a set of disks in the plane, and Q a set of points so that (a) no point from Q is contained in any disk from R, and (b) no face in the complement of the union of the disks in R contains more than one point of Q. Then $|R| = \Omega(|Q|)$.*

Proof. The number of faces in the in the complement of the union of the disks in R is $O(|R|)$. □

Covering vs. Separating. The input to our problem is a set I of n unit disks, and P a set of k points such that I separates P. Let $P_c \subseteq P$ denote those points contained in some disk in I; and P_s denote the remaining points. We compute an α-approximation to the smallest subset of I that covers P_c using a traditional set-cover algorithm; there are several poly-time algorithms that guarantee that $\alpha = O(1)$. We compute a β-approximation to the smallest subset of I that separates P_s, using the algorithm developed in the rest of this article. We argue below that the combination of the two solutions is an $O(\alpha + \beta)$ approximation to the smallest subset of I that separates P.

Let $OPT \subseteq I$ denote an optimal subset that separates P. Suppose that OPT covers k_1 of the points in P_c and let $k_2 = |P_c| - k_1$. By Lemma 1, $|OPT| = \Omega(k_2)$.

Now, by picking one disk to cover each of the k_2 points of P_c not covered by OPT, we see that there is a cover of P_c of size at most $|OPT| + k_2 = O(|OPT|)$. Thus, our α-approximation has size $O(\alpha) \cdot |OPT|$. Since OPT also separates P_s, our β-approximation has size $O(\beta) \cdot |OPT|$. Thus the combined solution has size $O(\alpha + \beta) \cdot |OPT|$.

In the rest of the article, we abuse notation and assume that no point in the input set P is contained in any disk in I, and describe a poly-time algorithm that computes an $O(1)$-approximation to the optimal subset of I that separates P.

3 Separating Two Points

Let s and t be two points in the plane, and G a set of disks such that no disk in G contains either s or t, but G separates s and t. See Figure 2. Our goal is to find the smallest cardinality subset B of G that separates s and t. We describe below a polynomial time algorithm that returns a constant factor approximation to this problem.

Without loss of generality, we may assume that the intersection graph of G is connected. (Otherwise, we apply the algorithm to each connected component for which the disks in the component separate s and t. We return the best solution obtained.) Let f_s and f_t denote the faces containing s and t, respectively, in the arrangement of G. We augment the intersection graph of G with vertices corresponding to s and t, and add an edge from s to each disk that contributes an edge to the boundary of the face f_s, and an edge from t to each disk that contributes an edge to the boundary of the face f_t. We assign a cost of 0 to s, t, and a cost of 1 to each disk in G. We then find the shortest path from s to t in
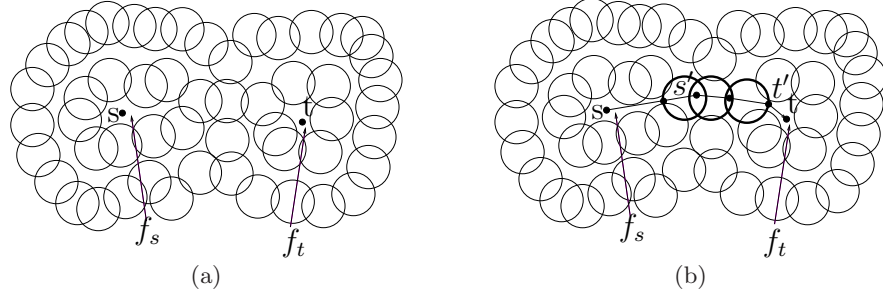
Fig. 2. (a) The figure shows faces f_s and f_t (b) This figure shows the sequence of disks in σ (their boundaries are bold) and the path π.

this graph, where the length of a path is the number of the *vertices* on it that correspond to disks in G. Let σ denote the sequence of disks on this shortest path. Note that any two disks that are not consecutive in σ do not intersect.

Using σ, we compute a path π in the plane, as described below, from s to t so that (a) there are points s' and t' on π so that the portion of π from s to s' is in f_s, and the portion from t' to t is in f_t; (b) every point on π from s' to t' is contained in some disk from σ; (c) the intersection of π with each disk in σ is connected. See Figure 2.

Suppose that the sequence of disks in σ is $d_1, \ldots, d_{|\sigma|}$. Let s' (resp. t') be a point in d_1 (resp. d_σ) that lies on the boundary of f_s (resp. f_t). For $1 \leq i \leq |\sigma|-1$, choose x_i to denote an arbitrary point in the intersection of d_i and d_{i+1}. The path π is constructed as follows: Take an arbitrary path from s to s' that lies within f_s, followed by the line segments $\overline{s'x_1}, \overline{x_1x_2}, \ldots, \overline{x_{|\sigma|-2}x_{|\sigma|-1}}, \overline{x_{|\sigma|-1}t'}$, followed by an arbitrary path from t' to t that lies within f_t.

Properties (a) and (b) hold for π by construction. Property (c) is seen to follow from the fact that disks that are not consecutive in σ do not overlap.

Notice that π "cuts" each disk in σ into two pieces. (Formally, the removal of π from any disk in σ yields two connected sets.) The path π may also intersect other disks and cut them into two or more pieces, and we refer to these pieces as disk pieces. For a disk that π does not intersect, there is only one disk piece, which is the disk itself.

We consider the intersection graph H of the disk pieces that come from disks in G. Observe that a disk piece does not have points on π, since π is removed; so two disk pieces intersecting means there is a point outside π that lies in both of them. In this graph, each disk piece has a cost of 1.

In this graph H, we compute, for each disk $d \in \sigma$, the shortest path between the two pieces corresponding to d. Suppose $d' \in \sigma$ yields the overall shortest path σ'; let D denote the set of disks that contribute a disk piece to this shortest path. Our algorithm returns D as its computed solution. See Figure 3.

We note that D separates s and t – in particular, the union of the disk pieces in σ' and the set $\pi \cap d'$ contains a cycle in the plane that intersects the path π between s and t exactly once.
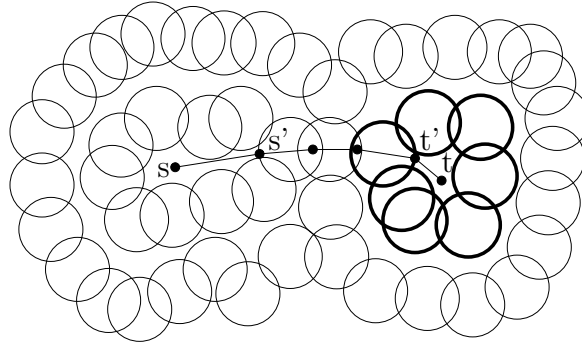
Fig. 3. This figure continues with the example of Figure 2. The disks with bold boundary are the set D computed by our algorithm. The only disk from G with bold boundary has two disk pieces, and the shortest path between them in graph H yields D.

3.1 Bounding the Size of the Output

Let B^* denote the smallest subset of G that separates s and t. We will show that $|D| = O(|B^*|)$. Let f^* denote the face containing s in the arrangement of B^*. Due to the optimality of B^*, we may assume that the boundary of f^* has only one component. Let a (resp. b) denote the first (resp. last) point on path π where π leaves f^*. It is possible that $a = b$. We find a minimum cardinality contiguous subsequence $\overline{\sigma}$ of σ that contains the subpath of π from a to b; let d_a and d_b denote the first and last disks in $\overline{\sigma}$. See Figure 4.
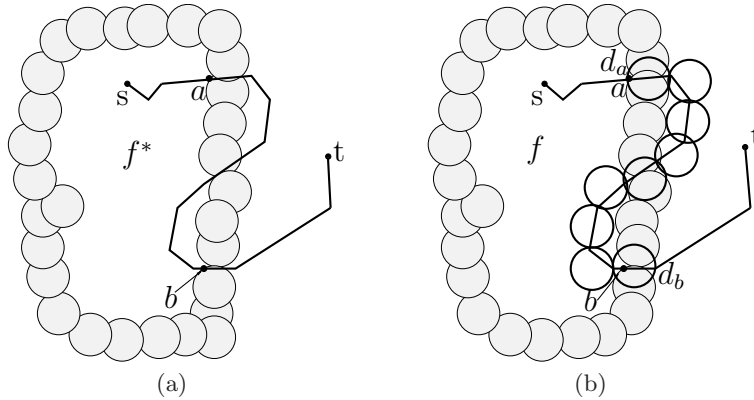

(a)

(b)

Fig. 4. The shaded disks are in B^*.(a) This figure shows points a and b where π leaves f^* for the first and last time, respectively. (b) This figure shows the face f containing s in the arrangement with $B^* \cup \overline{\sigma}$

We claim that $|\overline{\sigma}| \leq |B^*| + 2$; if this inequality does not hold, then we obtain a contradiction to the optimality of σ by replacing the disks in the $\overline{\sigma} \setminus \{d_a, d_b\}$ by B^*.

Consider the face f containing s in the arrangement with $B^* \cup \overline{\sigma}$. Each edge that bounds this face comes from a single disk piece, except for one edge corresponding to d_a that may come from two disk pieces. (This follows from the fact that the portion of π between a and b is covered by the disks in $\overline{\sigma}$.) These disk pieces induce a path in H in between the two pieces from d_a, and their cost therefore upper bounds the cost of D. We may bound the cost of these disk pieces by the number of edges on the boundary of f (with respect to $B^* \cup \overline{\sigma}$). The number of such edges is $O(|B^*| + |\overline{\sigma}|) = O(|B^*|)$.

Theorem 1. *Let s and t be two points in the plane, and G a set of disks such that no disk in G contains either s or t, but G separates s and t. There is a polynomial time algorithm that takes such G, s, and t as input, and outputs a subset $B \subseteq G$ that separates s and t; the size of B is at most a multiplicative constant of the size of the smallest subset $B^* \subseteq G$ that separates s and t.*

4 Separating Multiple Points

We now present a polynomial time algorithm that yields an $O(1)$ approximation to the problem of finding a minimum subset of I that separates the set P of points. The algorithm is obtained by calling recSep(P), where recSep(Q), for any $Q \subseteq P$ is the following recursive procedure:

1. If $|Q| \leq 1$, return $\emptyset$.
2. For every pair of points $s, t \in Q$, invoke the algorithm of Theorem 1 (with $G \leftarrow I$) to find a subset $B_{s,t} \subseteq I$ such that $B_{s,t}$ separates s and t.
3. Let B denote the minimum size subset $B_{s,t}$ over all pairs s and t considered.
4. Consider the partition of Q into subsets so that each subset corresponds to points in the same face (with respect to B). Suppose $Q_1, \ldots, Q_\tau$ are the subsets in this partition. Note that $\tau \geq 2$, since B separates some pair of points in Q.
5. Return $B \cup \bigcup_{j=1}^{\tau} \text{recSep}(Q_j)$.

Clearly, recSep(P) yields a separator for P. To bound the size of this separator, let us define a set $\mathcal{Q}$ that contains as its element any $Q \subseteq P$ such that $|Q| \geq 2$ and recSep(Q) is called somewhere within the call to recSep(P). For any $Q \in \mathcal{Q}$, define B_Q to be the set B that is computed in the body of the call to recSep(Q). Notice that recSep(P) returns $\cup_{Q \in \mathcal{Q}} B_Q$.

Now we "charge" each such B_Q to an arbitrary point within $p_Q \in Q$ in such a way that no point in P is charged more than once. A moment's thought reveals that this is indeed possible. (In a tree where each interval node has degree at least 2, the number of leaves is greater than the number of internal nodes.)

Let OPT denote the optimal separator for P and let $F_Q \subseteq OPT$ denote the disks that contribute to the boundary of the face (in the arrangement of OPT)

containing p_Q. We claim that $|B_Q| = O(|F_Q|)$; indeed F_Q separates $p_Q \in Q$ from any point in P, and thus any point in Q. Thus for any $t \in Q \setminus \{p_Q\}$, we have $|B_Q| \leq B_{p_Q,t} = O(|F_Q|)$.

We thus have

$$\bigcup_{Q \in \mathcal{Q}} |B_Q| \leq \sum_{Q \in \mathcal{Q}} O(|F_Q|) = O(|OPT|),$$

where the last equality follows from union complexity.

We have derived the main result of this paper:

Theorem 2. *Let I be a set of n unit disks and P a set of k points such that I separates P. There is a polynomial time algorithm that takes as input such I and P, and returns a subset $O \subseteq I$ of disks that also separates P, with the guarantee that $|O|$ is within a multiplicative $O(1)$ of the smallest subset of I that separates P.*

5 Conclusions

We have a presented an $O(1)$-approximation algorithm for finding the minimum subset of a given set I of disks that separates a given set of points P. One way to understand our contribution is as follows. Suppose we had at our disposal an efficient algorithm that optimally separates a single point $p \in P$ from every other point in P. Then applying this algorithm for each point in P, we get a separator for P. That the size of this separator is within $O(1)$ of the optimal is an easy consequence of union complexity. However, we only have at our disposal an efficient algorithm for a weaker task: that of approximately separating two given points in P. What we have shown is that even this suffices for the task of obtaining an $O(1)$ approximation to the overall problem.

It is easy to see that our algorithm and the approximation guarantee generalize, for example, to the case when the disks have arbitrary and different radii.

References

1. Pankaj K. Agarwal and Micha Sharir. *Davenport-Schinzel Sequences and Their Geometric Applications, Cambridge University Press.* 1998.
2. Boris Aronov, Esther Ezra, and Micha Sharir. Small-size epsilon-nets for axis-parallel rectangles and boxes. *SIAM J. Comput.*, 39(7):3248–3282, 2010.
3. Paul Balister, Zizhan Zheng, Santosh Kumar, and Prasun Sinha. Trap coverage: Allowing coverage holes of bounded diameter in wireless sensor networks. In *In Proc. of IEEE INFOCOM, Rio de Janeiro*, 2009.

4. Hervé Brönnimann and Michael T. Goodrich. Almost optimal set covers in finite vc-dimension. *Discrete & Computational Geometry*, 14(4):463–479, 1995.

5. Parinya Chalermsook and Julia Chuzhoy. Maximum independent set of rectangles. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, pages 892–901, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics.

6. Timothy M. Chan and Sariel Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. In *Symposium on Computational Geometry*, pages 333–340, 2009.

7. Kenneth L. Clarkson and Kasturi Varadarajan. Improved approximation algorithms for geometric set cover. In *Proceedings of the twenty-first annual symposium on Computational geometry*, SCG '05, pages 135–141, New York, NY, USA, 2005. ACM.

8. Jeff Erickson and Sariel Har-Peled. Optimally cutting a surface into a disk. *Discrete & Computational Geometry*, 31(1):37–59, 2004.

9. Jacob Fox and Janos Pach. Computing the independence number of intersection graphs. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, 2011.

10. Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Multiway cuts in node weighted graphs. *Journal of Algorithms*, 50(1):49 – 61, 2004.

11. Santosh Kumar, Ten H. Lai, and Anish Arora. Barrier coverage with wireless sensors. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 284–298, New York, NY, USA, 2005. ACM.

12. Nabil H. Mustafa and Saurabh Ray. Ptas for geometric hitting set problems via local search. In *Symposium on Computational Geometry*, pages 17–22, 2009.

13. John Reif. Minimum s-t cut of a planar undirected network in $o(n \log^2 n)$ time. *SIAM Journal on COmputing*, 12:71–81, 1983.

14. Swaminathan Sankararaman, Alon Efrat, Srinivasan Ramasubramanian, and Javad Taheri. Scheduling sensors for guaranteed sparse coverage. *CoRR*, abs/0911.4332, 2009.